

**State of Michigan**  
**Acme Interface Project**  
**Software Configuration Management Plan**

**General Information**

<i><b>System or Project ID / Acronym:</b></i>	Acme Interface Project	<i><b>Creation Date:</b></i>	1/20/2009
<i><b>Client Agency:</b></i>	State Budget Office	<i><b>Modification Date:</b></i>	1/20/2008
<i><b>Author(s):</b></i>	George Fictitious Configuration Management Manager	<i><b>MDIT Authorized by:</b></i>	John Pretend Office of Financial Management Director

**Privacy Information**

This document may contain information of a sensitive nature. This information should not be given to persons other than those who are involved in the **Acme Interface Project** project or who will become involved during the lifecycle.

## Table of Contents

General Information .....	1
Table of Contents .....	2
1. Introduction .....	3
1.1 Purpose .....	3
1.2 Objectives .....	3
1.3 References .....	4
2. Software Configuration Management Resources .....	4
2.1 Roles and Responsibilities .....	4
2.1.1 Project Manager (PM) .....	4
2.1.2 Configuration Management Manager (CMM) .....	4
2.1.3 Repository Administrator .....	5
2.1.4 Repository Gatekeeper .....	5
2.1.5 Librarian .....	5
2.1.6 Configuration Control Board (CCB) .....	5
2.1.7 Local Change Board (LCB) .....	6
2.1.8 Enterprise Change Board (ECB) .....	6
2.2 Resource Assignments .....	7
3. Software Configuration Management Tasks .....	8
3.1 Identification of Configuration Items .....	8
3.2 Configuration Items (CIs) .....	8
3.3 Baseline Identification .....	10
3.4 Repository Identification .....	10
3.5 Configuration Item Identifier .....	11
3.5.1 Configuration Identifier Process .....	11
3.5.2 Guidelines for Incrementing Release Version Numbers .....	12
3.5.3 Documentation Configuration Item Identifier .....	12
4. Software Configuration Control .....	13
4.1 Procedures for Changing Baselines .....	13
4.1.1 Code Baselines .....	13
4.1.2 Documentation Baselines .....	14
4.2 Change Requests and Approvals .....	14
4.2.1 Configuration Control Board Decision Process .....	14
4.2.2 Configuration Control Board Software Change Request Research .....	14
4.2.3 Configuration Control Board Prioritization .....	15
4.3 Change Control Process .....	15
4.4 Request for Change (RFC) Process .....	17
4.5 Level of Control .....	17
4.6 Management of Release Documentation .....	18
4.7 Configuration Control Tools and Techniques .....	18
5. Configuration Status Accounting .....	19
6. Configuration Audits and Reviews .....	19
6.1 Functional Configuration Audits .....	20
6.2 Physical Configuration Audits .....	20
7. Archive and Retrieval .....	20
8. Training .....	21
9. Related Forms .....	21
Approval Information .....	21

## **1. Introduction**

This Software Configuration Management (SCM) Plan specifically addresses configuration management for software. Configuration management for hardware, telecom, operating systems, and other components managed by Infrastructure Services are addressed by the MDIT Information Technology Infrastructure Library (ITIL) Process and Procedures.

### **1.1 Purpose**

The purpose of Software Configuration Management (SCM), in general, is to establish and maintain the integrity of work products using:

- Configuration Identification
- Configuration Control
- Configuration Status Accounting
- Configuration Audit

A Configuration Item (CI) is an entity designated for configuration management, which may consist of multiple related work products that form a baseline. This logical grouping provides ease of identification and controlled access. The selection of work products for configuration management should be based on criteria established during planning. Section 3 of this SCM Plan contains detailed information about CIs.

#### **Configuration Identification**

The purpose of Configuration Identification is to define the functional and physical characteristics of a CI in sufficient detail so that it may be developed, tested, evaluated, produced, competitively procured, accepted, operated, maintained, and supported. Configuration Identification is established by baselines plus approved changes. For purposes of this SCM Plan, Configuration Identification includes the selection, creation, and specification of the following:

- Products that are delivered to the client
- SEM documents requiring Structured Walkthroughs (SWT)

#### **Configuration Control**

The process of evaluating, approving or disapproving, and managing changes to controlled items. This includes tracking the configuration of each of the CIs, approving a new configuration if necessary, and updating the baseline.

#### **Configuration Status Accounting**

The process of creating and organizing the information necessary for the performance of configuration management. An element of configuration management consisting of the recording and reporting of information needed to manage a configuration effectively. This information includes a listing of the approved configuration identification, the status of proposed changes to the configuration, and the implementation status of approved changes.

#### **Configuration Audit**

Audits are conducted to verify that a CI, or a collection of CIs that make up a baseline, conforms to a specified standard or requirement. This includes functional and physical configuration audits.

### **1.2 Objectives**

This SCM Plan defines the configuration management policies and procedures for the Acme Interface Project. This plan has been developed early in the lifecycle to ensure the control of changes as soon as the project requirements are approved. This plan has been developed early in the lifecycle to ensure the control of changes as soon as the project requirements are approved and base lined. This plan addresses activities that are platform independent, such as identifying the items that will be placed under configuration management. As the project progresses through the lifecycle stages, the plan is expanded to reflect platform specific activities.

Changes in this system affecting other SCM plans are identified and explained in Section 2 (Software Configuration Management Resources) and Section 3 (Software Configuration Management Tasks) of this plan.

## **1.3 References**

Listed here are policies, procedures and standards used in preparing and setting up this SCM Plan.

- State of Michigan's System Engineering Methodology (SEM)
- Request for Quote #123-08 Statewide Web Purchasing System
- Vendor Proposal for Statewide Web purchasing System

## **2. Software Configuration Management Resources**

This section identifies the roles of individuals and groups that participate in the SCM process. It describes the relationships between individuals and groups.

### **2.1 Roles and Responsibilities**

Only the responsibilities related to SCM are listed here.

#### **2.1.1 Project Manager (PM)**

- Establish the overall project schedule for SCM activities with Configuration Management Manager (CMM)
- Make sure team members are knowledgeable of SCM concepts and techniques and that they are applied to project activities
- Ensure compliance with the SCM standards and procedures set by the CMM, the Configuration Control Board (CCB), and any other affected groups as outlined in this plan

#### **2.1.2 Configuration Management Manager (CMM)**

The project CMM will prepare the SCM Plan with assistance from the Project Manager. The CMM is responsible for creating and/or updating the SCM Plan, as well as communicating the contents of the plan to the project team.

### **Responsibilities**

#### **SCM Planning**

- Identify the Configuration Items (CIs) to be managed under the SCM processes
- Create, manage and maintain the SCM Plan, standards, and procedures
- Communicate any changes to the SCM Plan, standards, and procedures to all stakeholders
- Make sure that all project team members involved in the SCM process receive training on their roles
- Make updates to the SCM Plan, as appropriate

- Make sure that any updates to the SCM Plan are communicated to the appropriate project team members
- Form and lead a SCM Team
- Approve changes to the SCM Plan

### **Implementing Changes**

- Participate as a member of the Configuration Control Board (CCB)
- Create SCM products (baselines, application environments), as authorized by the CCB
- Process and track software change requests
- Function as the point of contact with Infrastructure Services to analyze proposed changes and to insure interoperability between hardware and software components

### **Tracking, Reporting and Audits**

- Make sure that configuration item change requests and problem reports for all CIs are initiated, recorded, reviewed, approved, and tracked according to the SCM Plan
- Ensure all Functional and Physical Configuration Audits are performed
- Respond to requests for status regarding SCM activities from managers and auditors

#### **2.1.3 Repository Administrator**

##### **Responsibilities**

- Create and manage repository work areas and containers
- Grant and manage access to all work areas and containers
- Create and distribute processes, procedures, and standards for the Software Configuration Management repository
- Provide technical assistance to project staff using the repository
- Enforce repository naming standards

#### **2.1.4 Repository Gatekeeper**

The Repository Gatekeeper is responsible for the initial check-in of the configuration items. The project manager and the manager for each of the development groups, including the DBA manager, perform these tasks for their respective areas.

- Ensure documentation deliverables meet repository naming standards
- Check in new or modified documents
- Ensure all repository procedures are followed

#### **2.1.5 Librarian**

- Label baselines
- Promote CIs in accordance with processes set forth in this plan
- Assemble configuration reports
- Participate in configuration audits

#### **2.1.6 Configuration Control Board (CCB)**

##### **Responsibilities**

- Monitor changes and updates to project requirements
- Authorize the establishment of baselines and the identification of CIs
- Ensure that all approved changes and updates to CIs are placed under configuration control

- Use the SCM Plan as its primary decision-making resource
- Support and provide input to Local Change Board (LCB) and Enterprise Change Board (ECB) functions related to the MDIT Service Management Center Request for Change (RFC) process
- Review and authorize changes to the baselines
- Attend regularly scheduled meetings
- Review and discuss new change requests
- Prioritize change requests
- Authorize research on change requests
- Approve the commencement of work on change requests (make active)
- Review the status of active change requests
- Create and communicate minutes from the CCB to affected groups

## Roles

Members	Roles
Director of the Office of Financial Management (OFM)	System Owner with decision making authority
Manager of OFM Purchase Accounting	Stakeholder
Manager of OFM Auditing	Stakeholder
Manager of OFM GAAP Accounting	Stakeholder
MDIT Project Manager	Project Manager for the Acme Interface Project
MDIT AS DMB Client Service Director (CSD)	Stakeholder
MDIT AS SFS Application Development Manager	Development Manager
MDIT AS SFS Data Warehouse Manager	Development Manager
MDIT AS Technical Operations Manager	Technical Manager
Configuration Management Manager	Service Provider
Librarian	Service Provider

### 2.1.7 Local Change Board (LCB)

## Responsibilities

- Authorize changes to the system
- Verify that any changes with statewide impact are marked for Enterprise Change Board (ECB) approval

## Roles

Members	Roles
MDIT AS DMB Client Service Director (CSD)	Stakeholder
MDIT Application Development Managers	Development Manager
MDIT AS SFS Data Warehouse Manager	Development Manager
MDIT AS Technical Operations Manager	Technical Manager
Client Support Specialist	Client Support
Infrastructure Specialist	MDIT Agency Service Support
Configuration Management Manager	Service Provider

### 2.1.8 Enterprise Change Board (ECB)

## Responsibilities

- Ensure changes do not adversely affect other systems
- Authorize changes to the systems

## Roles

The ECB is primarily staffed with MDIT Infrastructure representatives. Attendance at ECB meetings by the local staff will vary depending on the scope of the change. Typically only one or two of the following will attend.

Members	Roles
MDIT AS DMB Client Service Director (CSD)	Stakeholder
MDIT Application Development Managers	Development Manager
MDIT AS SFS Data Warehouse Manager	Development Manager
MDIT AS Technical Operations Manager	Technical Manager
Client Support Specialist	Client Support
Infrastructure Specialist	MDIT Agency Service Support
Configuration Management Manager	Service Provider

## 2.2 Resource Assignments

SCM Role Assignments		System Owner	Project Management	Config Management Manager	MDIT Application Development Manager	CCB Facilitator	Repository Gate Keeper	Repository Administrator	Librarian
<b>KEY</b> <b>P</b> - Primary Role <b>R</b> - Role Responsibilities									
John Pretend, Director,	Office of Financial Mgmt	P							
Tom McBlieve,	Project Manager		P						
George Fictitious,	Technical Operations Support			P		P	R		
Bill Meenger,	Application Development Manager				P		R		
Carol Joyner,	Database Manager						R		
Linda Binary,	Technical Operations Support							P	
Tom Filer,	Technical Operations Support								P

**Table 1 Resource Assignments**

### 3. Software Configuration Management Tasks

This section consists of the following:

- Identification of Configuration Items
- Configuration Items
- Baseline Identification
- Repository Identification
- Configuration Item Identifier

#### 3.1 Identification of Configuration Items

The terms Configuration Identification and Configuration Item are defined in Section 1.1 of this document.

In this SCM Plan, work products are considered for configuration management based on the following criteria. A work product is any tangible item that results from a project function, activity or task.

- May be used by one or more work groups
- Are expected to change over time either because of errors or change of requirements
- Are dependent on each other in that a change in one mandates a change in another/others
- Are critical to the project

Items in the following categories are selected to be placed under configuration management:

- Project Management documentation, including Project Plan and Project Charter
- SEM documentation, including all deliverables, Structured Walkthroughs (SWT), Stage Exit Position Response form
- Models
- Interfaces
- Process descriptions
- Product/Application data such as lookup tables, system files
- Source code and executable code
- Test scripts
- Test data
- Metrics, status reports, quality review reports, etc.
- Support tools, including compilers, editors, testing tools
- Touch Point documentation including EA solution documents, Infrastructure Services Request (DIT-0184), and Security Plan and Assessment (DIT-0170)

#### 3.2 Configuration Items (CIs)

The following table contains CIs that are included in this SCM Plan.

Configuration Items	Description/SUITE Template	Responsible for placing item under control	When item is put under control
Project Charter	PMM-02	Project Manager	Initiation & Planning Stage Exit
Project Plan	PMM-03	Project Manager	Initiation & Planning Stage Exit
Security Plan	DIT-0170	OES Liaison	Initiation & Planning Stage Exit
Software Configuration	SEM-0302	CM Manager	Initiation & Planning Stage



<b>Configuration Items</b>	<b>Description/SUITE Template</b>	<b>Responsible for placing item under control</b>	<b>When item is put under control</b>
Management Plan			Exit
Maintenance Plan	SEM-0301	MDIT Analyst/CM Manager	Initiation & Planning Stage Exit
Software Change Request	DIT-0178	Business Owner/Project Manager/MDIT Analyst	Initiation & Planning Stage Exit
Software Change Control Log	DIT-0179	Business Owner/Project Manager/MDIT Analyst	Initiation & Planning Stage Exit
Requirements Specification	SEM-0402	Business Owner	Requirements Stage Exit
Requirements Traceability Matrix	SEM-0401	Project Manager/MDIT Analyst	Requirements Stage Exit
Requirements Management Checklist	SEM-0403	Project Manager/MDIT Analyst	Requirements Stage Exit
EA Solutions Assessment	SEM Touch Point	Project Manager/MDIT Analyst	Requirements Stage Exit
Infrastructure Services Request (ISR)	SEM Touch Point, DIT-0184	Project Manager/MDIT Analyst	Requirements Stage Exit
Hosting Solution	SEM Touch Point	Project Manager/MDIT Analyst	Requirements Stage Exit
Functional Design	SEM-0501	Business Owner /Project Manager/MDIT Analyst	Functional Design Stage Exit
Conversion Plan	SEM-0601	Project Manager/MDIT Analyst	System Design Stage Exit
Test Plan	SEM-0602	Business Owner /Test Manager	System Design Stage Exit
Test Reports	SEM-0603	Business Owner/Test Manager	System Design Stage Exit
System Design	SEM-0604	Project Manager/MDIT Analyst	System Design Stage Exit
System Design Checklist	SEM-0605	Project Manager/MDIT Analyst	System Design Stage Exit
Software Testing Checklist	SEM-0606	Project Manager/MDIT Analyst	System Design Stage Exit
Transition Plan	SEM-0701	Project Manager/MDIT Analyst	Construction Stage Exit
Installation Plan	SEM-0702	Project Manager/MDIT Analyst	Construction Stage Exit
Training Plan	SEM-0703	Business Owner	Construction Stage Exit
Training Plan checklist	SEM-0704	Business Owner	Construction Stage Exit
Testing Package	SEM-0801, SEM-0802, SEM-0803, SEM-0804, and SEM-0805	Business Owner/Project Manager/MDIT Analyst	Testing Stage Exit
Release Notes	Word/Excel	Project Manager/MDIT Analyst	Implementation Stage Exit
Post Implementation Evaluation Report	PMM-16	Business Owner/Project Manager/MDIT Analyst	Implementation Stage Exit
Request for Change (RFC)	SEM Touchpoint, SMC website	Project Manager/MDIT Analyst	Construction Stage Exit
Structured Walkthrough Meeting Record	SEM-0187	Business Owner/Project Manager/MDIT Analyst	All Stages
Structured Walkthrough Summary Record	SEM-0188	Business Owner/Project Manager/MDIT Analyst	All Stages
Stage Exit Position Response	SEM-0189	Business Owner/Project Manager/MDIT Analyst	All Stages
Project Charter	PMM-02 EXP	Project Manager	Initiation, Requirements & Design Stage Exit
Project Plan	PMM-03 EXP	Project Manager	Initiation, Requirements & Design Stage Exit
Initiation, Requirements and Design Plan	SEM EXP-01	Business Owner/Project Manager/MDIT Analyst	Construction & Testing Stage Exit

Configuration Items	Description/SUITE Template	Responsible for placing item under control	When item is put under control
Construction and Testing Plan	SEM EXP-02	Business Owner/Project Manager/MDIT Analyst	Construction & Testing Stage Exit
Post Implementation Evaluation Report	PMM-16 EXP	Business Owner/Project Manager/MDIT Analyst	Implementation Stage Exit
C/JAVA Code (Example)	Application Source Code	Developer	Initial unit test
Database Stored Procedures	Database Source Code	DBA	Initial unit test
Cobol Compiler (Example) File Editor (Example)	Support Tools	Infrastructure	After received from vendor
Graphics/Images	User Interface Elements	Graphic Designer	Initial unit test

### 3.3 Baseline Identification

This application consists of four environments: Development (DEV), System Test (ST), User Acceptance Testing (UAT), and Production (PRD).

Baselines are created for three of the environments: ST, UAT, and PRD. Developers have an Integrated Development Environment (IDE) installed on their desktop computers. The IDE has the capability of hosting the application. By not creating a development baseline the developers have more flexibility for research and development.

A baseline is created whenever changes are made to any of the three remaining environments. The COTS software used for repository functions has a feature called Label that automates this process.

### 3.4 Repository Identification

There are two types of repositories used to store configuration items, an electronic repository and a physical (filing cabinet) repository. The electronic repository is used to store computer generated files such as source code and documentation. The physical repository stores other documents that do not have an electronic version or documents containing signatures.

#### Electronic Repository:

Serena Version Manager is used for versioning of electronic files in accordance with processes set forth in this plan. MDIT Infrastructure Services provides the storage location for the files. The server is HCS084TESTPA001, the share is named MDIT2, and the folder location on that share is \AgyServ-DMB\shared\SFS\repository. Backups of the repository are done in accordance with the standard operating procedure for server backups.

#### Physical Repository:

##### Paper documents

Documents that cannot be stored electronically are maintained by the division's administrative staff. Examples of files that cannot be stored electronically are contracts, signature pages, and correspondence received by standard mail. A filing cabinet adjacent to the division's administrative assistant is used. This plan outlines a procedure for recording and submitting the documents to the administrative staff for storage.

##### Electronic media (CD, DVD, Flash drives, tapes etc)

COTS software packages used either in the construction or operation of the system are configuration items and need to be tracked. MDIT Office Automation process and procedures are followed. A configuration item record is created in **Error! Reference source not found.** for each COTS package documenting its name and version number. Office Automation is in charge of keeping the media and performing the installations.

Electronic media not containing COTS software needs to be reviewed by the CCB to see if it meets the criteria for configuration items. If it meets the criteria then it must follow the same procedures outlined for paper documents.

### 3.5 Configuration Item Identifier

Configuration Item Identifiers are used to label all of the CIs that make up a particular grouping such as an application release, a project development phase or documentation changes.

This identification scheme preserves all of the files that are used to create each release and exactly which versions of those files were used. This scheme works for the application installations and then for subsequent upgrades.

Identifiers are used to label the documentation deliverables in a project. For instance, at the end of the system design stage, all of the approved deliverables will be labeled and preserved for future reference. After the completion of the project, many of the deliverables will need to be updated to reflect changes to the application. Those deliverables are assigned identification labels so that their current state can be identified and preserved for future reference.

#### 3.5.1 Configuration Identifier Process

The following describes how identifiers are assigned to software components.

The software configuration item repository is controlled using the COTS tool called Serena Version Manager. The repository consists of one Serena project database named SFSwebInterface. The Label feature of Serena Version Manager is used to assign the configuration identifier.

Application Environments:

Development	Exists on each developer's desktop computer and is integrated into the Integrated Development Environment for Visual Basic .Net.
System Test	Exists on the test server. It is used for MDIT to test the total application. Once testing is completed the code is moved to User Acceptance Testing.
User Acceptance Testing	Exists on the test server. This is where the final testing of the application is performed before it is moved to production.
Production	Exists on the production server. This is the final destination of the program code.

Except for the development environment, each time an environment changes it is considered a release and needs to be uniquely identified. The files identified in the release are referred to as the baseline. The development environment does not have its releases identified or baselined as each developer has a development environment on their desktop computer.

Serena Version Manager assigns a unique revision number as files are added and checked-in to the repository.

Once all of the files requiring change are checked-in, the MDIT development manager approves and the files are promoted to the System Test application environment. At the time of deployment, all of the files in the System Test environment are considered a released baseline and are assigned a configuration identifier. This identifier has a text portion and a numerical portion. The text portion identifies the application and environment and the numerical portion is a sequence number used to track changes. The numerical part is referred to as the version.

This table shows the software release configuration identification labeling scheme:

Environment	Identification Scheme
Development	N/A
System Test	SFSweb.TST.BL.mm.nn.xx
User Acceptance Testing	SFSweb.UAT.BL.mm.nn.xx
Production	SFSweb.PRD.BL.mm.nn.xx

Note: Where mm is Major, nn is Minor, and XX is Sequence.

Configuration identification labels are in the form of SFSweb.PRD.BL.mm.nn.xx where SFSweb stands for Statewide Financial System Web application, PRD is for production, BL stands for Baseline, and mm.nn.rr is the release version number.

For example the file global.asa is checked into the repository 3 times before it is ready for system test. This gives global.asa the revision number of 1.3. Once all files needing changes are checked into the repository, they are given the identifier SFSweb.TST.BL.01.00.00 and the files are migrated to the System Test environment.

Later a new feature is added to the application causing global.asa to change. Upon check in global.asa automatically receives a revision number of 1.4. All files in the application are given the baseline identifier of SFSweb.TST.BL.01.00.01 and any changed files are moved to the System Test environment.

Any references to version label SFSweb.TST.BL.01.00.00 access revision 1.3 of global.asa. Any references to SFSweb.TST.BL.01.00.01 access revision 1.4 of global.asa.

Two of the benefits of doing this are the ability to roll back to a previous production version and the ability to audit changes.

When the files are approved to be moved from System Test to User Acceptance Testing, the process repeats but with the identifier used for the UAT environment SFSweb.UAT.BL.01.00.00. This process then repeats for the move from UAT to production.

### **3.5.2 Guidelines for Incrementing Release Version Numbers**

Version Numbers are of the form Major.Minor.Sequence.

For most releases and bug fixes, the Sequence number is the only portion of the version number that is incremented. For example, version SFS.TST.BL.1.0.21 goes to 1.0.22.

The Minor number is incremented when a significant enhancement is implemented or if the Sequence number approaches a very large number. For example, if the Sequence number approaches 99, it is time to increment the Minor number. When the Minor number is incremented, the Sequence number resets to zero. For example, SFS.TST.BL.2.1.99 is incremented to SFS.TST.BL.2.2.0.

The Major number is rarely incremented. However, if the Minor number approaches 99, the Major number is incremented.

For example, SFS.TST.BL.2.99.99 would change to SFS.TST.BL.3.0.0 for the next System Test release. Major changes that impact most modules are good reasons to increment a Major number. When the Major number is incremented, the Minor and Sequence numbers are reset to zero.

The Configuration Control Board (CCB) may decide to reset the numbering to emphasize the changes in a particular release. For example, if the current release is SFS.TST.BL.01.03.15 and significant changes are made to the application then the release may be renumbered to SFS.TST.BL.01.04.00 or SFS.TST.BL.02.00.00. This decision is documented with the change request.

### **3.5.3 Documentation Configuration Item Identifier**

SEM documents are placed under configuration control. A separate folder called Documentation under the Serena project database SFSwebInterface stores the SEM documentation of this project.

A baseline of the documentation is taken at the end of each SEM project stage. All documentation receives the label whether it was modified or not. This way the current state of all the documentation can be retrieved for that stage showing a complete snapshot in time.

<b>Documentation Identification Scheme</b>
SFSweb.doc.BL.ss.xxxx

Note: Where ss is SEM stage abbreviation, and xxxx is a sequence number.

#### SEM Stage Abbreviation

Stage	Origin	Abbreviation
Initiation and Planning	SEM	IP
Requirements Definition	SEM	RD
Functional Design	SEM	FD
System Design	SEM	SD
Construction	SEM	CS
Testing	SEM	TS
Implementation	SEM	IS
Initiation, Requirements and Design	SEM Express	ID
Construction and Testing	SEM Express	CT

Table 2 SEM Stage Abbreviations

At the end of the Initiation and Planning stage, when all documentation has been checked into the repository, a configuration identifier is assigned. See **Error! Reference source not found.** for the list of documents requiring check-in at the end of each stage. The label feature of Serena Version Manager is used to assign the label SFSweb.doc.BL.IP.0001 to all documents.

Once the Requirements Definition stage is complete, all required documents are checked in and assigned the label of SFSweb.doc.BL.RD.0001. This process labels documents that have changed or are new. It also labels those that have not changed, such as the Project Charter and the Project Plan. This continues through the Implementation stage (SFSweb.doc.BL.IS.0001).

After the application is in production and maintenance is required, the appropriate documents are checked-out, modified, checked-in, and labeled. For example, at the end of the Construction stage, the first time maintenance is performed on this system the documentation is labeled SFSweb.doc.BL.CS.0002. Even though the requirements have not been modified since the end of the Design stage, they are labeled with the Construction stage label. If in the future there is a need to examine the work as it existed at the end of this Construction stage, the correct version of the requirements is easily identified.

## 4. Software Configuration Control

The term Configuration Control is defined in Section 1.1 of this document. Software configuration control includes the following objectives:

- Procedures for changing baselines
- Change requests approvals
- Responsibilities for change control
- Change control process
- Request for Change process
- Level of control
- Management of release documentation
- Configuration control tools and techniques

### 4.1 Procedures for Changing Baselines

#### 4.1.1 Code Baselines

Code baselines are established whenever a change is authorized for one of the environments. The three environments that have baselines maintained are System Test, User Acceptance Testing, and Production. The steps below are a high level process for changing the baseline.

1. The Software Change Request (SCR) status is updated to indicate that the CCB has approved its migration and which environment is to be updated. The librarian is notified to continue processing the SCR.
2. The librarian duties include:
  - 2.1. Use the label feature to assign a baseline label as described in the Configuration Identifier Process
  - 2.2. Stage the files listed in the SCR to stage area
  - 2.3. Assign the SCR to Technical Operations
3. Technical Operations duties include:
  - 3.1. Migrate the files to the appropriate environment
  - 3.2. Update the SCR
  - 3.3. Notify the CCB that the SCR may be closed

#### **4.1.2 Documentation Baselines**

Documentation baselines are created at the completion of each SEM stage. The librarian uses the signed Stage Exit form as a checklist to verify that all documents are checked in to the repository. The librarian then follows the document baseline naming convention outlined above to label all the documents with the current baseline label.

### **4.2 Change Requests and Approvals**

The COTS application Project Tracker is used for change management. All members of the project teams, OFM, helpdesk, local project management and MDIT employees assigned to this application have access to Project Tracker. Project Tracker includes a Software Change Request tracking system. Software Change Requests are used to report problems, identify new or changed requirements, and log suggestions for improvements.

Decisions for making changes to the application are made by the Configuration Control Board (CCB). The roles, responsibilities and members of the CCB are listed above in the Roles and Responsibilities section. The CCB prioritizes change requests, assigns them for work and tracks them to completion.

#### **4.2.1 Configuration Control Board Decision Process**

Decisions on change requests or the Software Configuration Management Plan are made on a consensus basis. If any of the participants have a strong objection to the decision, reasonable attempts are made to address those objections.

As the system owner, the Director of the Office of Financial Management (or designate) has final decision making authority. The director considers input from all meeting participants. Decisions that conflict with MDIT standards or policies cannot proceed until an exception is granted by using the MDIT exception process.

#### **4.2.2 Configuration Control Board Software Change Request Research**

This section defines basic elements of the research required for each software change request.

- The research on each software change request includes: (new)
- Reason for the request (legislative, State/Federal mandate, audit findings, security)
- Impact to the system
- Impact to related systems
- Estimated schedule
- Estimated cost
- Anticipated benefit

### 4.2.3 Configuration Control Board Prioritization

Members of the board primarily use their skill, knowledge, and experience to set priorities.

Some items to be considered in setting priorities are:

- Legislative mandates
- Citizen impact
- Cost saving
- Security

## 4.3 Change Control Process

This section contains both a simplified and expanded description of the process. (Note that the numbering scheme for the simple and expanded descriptions does not align.)

**Simplified Process Description** (Assumes approvals are granted at each step.)

1. A change is requested in the form of a Software Change Request (SCR).
2. SCR is presented to CCB for approval to perform research.
3. Researcher is assigned.
4. SCR with research and recommendation is presented to CCB.
5. CCB approves commencement of work.
6. Work is performed (programs changed, documentation changed).
7. CCB is notified that changes are ready for system test.
8. Developers perform system test.
9. CCB is notified that system test is successful and code is ready for User Acceptance Test (UAT); approves establishment of a system test baseline.
10. CCB approves the migration of the release to UAT and the establishment of a UAT baseline.
11. CCB sets a date for the release to be moved to production and authorizes the creation of a Request For Change (RFC).
12. CCB reviews changes ready for production.
13. LCB/ECB reviews changes ready for production.
14. Changes are moved to production.
15. Change request is updated and closed.

**Expanded Process Description**

1. A change is requested in the form of a Software Change Request (SCR).
  - a. A SCR is created and its status is set to open.
  - b. A SCR owner is assigned.
2. SCR is presented to CCB by the SCR owner for approval to perform research.
  - a. The CCB indicates any concerns and the owner adds those concerns to the SCR.
  - b. The CCB indicates if the standard research criterion is sufficient. If not, the owner updates the SCR with the modified research criteria.
  - c. If the SCR is not approved for research it is either closed or put on hold. Software Change Requests on hold are re-evaluated on a regular basis to be worked on, placed back on hold, or closed.
3. Researcher is assigned.
  - a. Either an MDIT developer or an Office of Financial Management (OFM) employee is assigned the SCR.
  - b. The researcher contacts the resource(s) needed to complete the research.
4. SCR with research and recommendation is presented to CCB.
5. CCB approves commencement of work.
  - a. If the SCR is not approved, it is returned for more research, closed, or put on hold.
6. The SCR is assigned to a MDIT developer to:
  - a. Follow the SEM procedures.
  - b. Update the SCR(s) with the file names of any configuration items to be modified.

- c. Check those configuration items out of the repository.
  - d. Modify the programs and test them on their desktop development environment.
  - e. Check the files into the repository.
  - f. Notify the MDIT AS Development Manager that the change is ready for system test.
7. The CCB is notified by the MDIT AS Development Manager that changes are ready for system test. Based on CCB approval, the CMM coordinates this change with any others changes that are ready for system test.
    - a. The CMM assigns the SCR(s) to the librarian.
    - b. The librarian retrieves the lists of configuration items from the SCR, extracts the files to the staging area, and updates and assigns the SCR to Technical Operations.
    - c. Technical Operations builds the new system test environment, updates the SCR(s) and assigns the SCR(s) to the development manager to initiate testing.
  8. Developers perform system test.
    - a. The development team performs testing according to the SEM Test Plan (DIT xxx) against the system test environment.
    - b. The process of coding, testing and rebuilding repeats until all test plans are successful.
  9. The development manager notifies the CCB that system test is successful.
    - a. The CCB approves establishment of a system test baseline.
    - b. The CMM notifies the librarian to label the system test configuration items with the new baseline label.
    - c. The librarian produces the System Test Baseline Content Report containing a list of all the CIs and their revision numbers.
    - d. The CMM compares System Test Baseline Content report to the SCR(s) and the Requirements Traceability Matrix. Note that this activity is known as a Functional Audit. See section 6.1. If any discrepancies are found, the CCB is notified and the SCR(s) are assigned to the development manager.
  10. CCB approves release to UAT.
    - a. CCB reviews all changes ready for UAT.
    - b. A list of changes to be migrated is determined and documented as the next release.
    - c. A functional audit is performed by comparing the Requirements Traceability Matrix to the list of changes for this release.
    - d. Approves the migration of the release to UAT.
    - e. Determines the date for the release to UAT.
    - f. Establish a UAT baseline.
    - g. Determines the UAT baseline label.
    - h. Establishes a date for production release.
  11. Based on CCB approval, the CMM performs the following:
    - a. Updates the SCR(s) with baseline label, the dates for move to UAT and the move to production.
    - b. Determines if this change has an enterprise wide effect.
    - c. Creates a Request for Change (RFC) in the MDIT SMC website documenting the move to production and noting if Enterprise Control Board approval is required.
    - d. Assigns the SCR(s) to the librarian.
  12. To prepare for the release to UAT:
    - a. The librarian labels the UAT files with the new baseline label.
    - b. The librarian produces the UAT Baseline Content Report containing a list of all the CIs in that baseline and their revision number.
    - c. The CMM compares UAT Baseline Content Report to the SCR(s) and the Requirements Traceability Matrix. Note that this activity is known as a Functional Audit. See section 6.1. If any discrepancies are found, the CCB is notified and the SCR(s) are assigned to the development manager.
    - d. The librarian extracts the files to the staging area.
  13. Technical Operations builds the release and promotes the code to the new UAT environment, updates and assigns the SCR(s) to the SCR owners.
  14. The Test Plan is executed in the UAT environment.
  15. If unsuccessful, the CCB is notified and determines the disposition of the release.
  16. Possible dispositions are:
    - a. Halting the release and returning it to development for resolution.
    - b. Removing some change requests from the release and move the release to production.
    - c. Accept the anomalies.
  17. Once the changes are successfully tested the CCB meets to review the results and:



- a. Approves/Disapproves scheduled changes.
  - b. Determines if any changes are needed to the RFC, and directs the CMM to make changes if necessary.
  - c. Updates the SCR to reflect the approval.
18. LCB/ECB meets to review changes ready for production and approves or rejects the move to production.
19. The librarian:
- a. Using the UAT Baseline Content Report, creates a new production baseline.
  - b. Creates the Production Baseline Content Report containing the baseline name and a list of all the CIs in that baseline and their revision number. Comparison of the UAT and Production Baseline Content Reports is known as a Physical Audit. See section 6.2.
  - c. The CMM compares Production Baseline Content Report to the SCR(s) and the Requirements Traceability Matrix. This activity is known as a Functional Audit. See section 6.1. If any discrepancies are found, the CCB is notified and the SCR(s) are assigned to the development manager.
  - d. Extracts documented files to the staging area.
  - e. Assigns the SCR(s) to the Technical Operations group.
20. Technical Operations builds the release and promotes the code to the production environment, updates the SCR(s), and assigns them to the SCR owner.
21. Application is validated per processes described in the Installation Plan (SEM-0702).
22. The SCR owner closes the SCR.
23. The CMM closes the RFC.

#### 4.4 Request for Change (RFC) Process

This Software Configuration Management (SCM) Plan supports the MDIT Request for Change (RFC) process. The Configuration Management Manager is responsible for ensuring that all LCB and ECB processes are followed.

An RFC is created when the Configuration Control Board approves a release to the UAT environment. The UAT environment follows the same two-week batch cycle as production. This allows for a full system test of the changes.

When the UAT environment successfully completes its two-week batch testing, the changes are moved to production. The RFC is created as soon as the release is approved for UAT because the batch testing is two weeks. This allows for the LCB to review the change in the first week and the ECB to review in the second week.

There are two meetings of the CCB in this cycle giving sufficient time for the approval process to complete.

#### 4.5 Level of Control

Different levels of control and their actions are assigned as follows:

##### Development Environment

Action	Control
Check in source code	Developers, librarians
Promote source code	Automatically when checked in
Check in project documentation	Project manager, developers, librarians
Check in application documentation	Project manager, developers, librarians

##### System Test Environment

Action	Control
Check in source code	Check in to system test not allowed
Authorize code promotion to system test	CCB
Promote source code to system test	Librarian
Define release	CCB
Authorize baseline label	CCB

Label baseline	Librarian
Authorize environment build	CCB
Build environment	Technical Operations

#### User Acceptance Test Environment

Action	Control
Check in source code	Check in to UAT not allowed.
Authorize code promotion to UAT	CCB
Promote source code to UAT	Librarian
Define release	CCB
Authorize baseline label	CCB
Label baseline	Librarian
Authorize environment build	CCB
Build and release environment	Technical Operations

#### Production Environment

Action	Control
Check in source code	Check in to production not allowed.
Authorize code promotion to Production	CCB
Promote source code to Production	Librarian
Define release	CCB
Authorize baseline label	CCB
Label baseline	Librarian
Authorize environment build	CCB
Build and release environment	Technical Operations

## 4.6 Management of Release Documentation

The SUITE Installation Plan (SEM-0702) contains enough detail to serve as Release Management for this project. A template of the Installation Plan is filled out for each release. The Project Management Manager (PMM) is responsible to ensure this is completed. The initial version of the Installation Plan is completed before the migration to the UAT environment allowing it to be tested before the production installation. The plan is finalized at the end of the SEM Testing Stage. A section in the Installation Plan contains a plan for backing out the release.

#### User Documentation

Office of Financial Management maintains a communication group that prepares release notes for the end users describing any changes to the system.

The PMM is responsible for ensuring document deliverables are put under configuration control. **Error! Reference source not found.** contains a list of the deliverables.

## 4.7 Configuration Control Tools and Techniques

Serena Version Manager is used for versioning of electronic files as described in this plan. The storage location for the files is supplied by MDIT Infrastructure Services. The server is HCS084TESTPA001 and the share is named MDIT2 and the folder location on that share is \AgyServ-DMB\shared\SFS\repository. Backups of the repository are done in accordance with the standard operating procedure for servers.

The current backup procedure is full system backups on the weekend and differential backups each weekday. The backup files are stored for three month.

The application Project Tracker is used for change control. All members of the projects team, OFM, helpdesk, project management and MDIT employees have access to the system. They enter requests for change in the portion of the

system called Software Change Requests. The Service Ticket portion of the system is used to enter problems and incidents. This is an established application housed at the hosting center, administered by business analysts and local MDIT operations staff. Backups of this system are done in accordance with the standard operating procedure for servers.

## **5. Configuration Status Accounting**

Configuration Status Accounting has the following objectives in this Software Configuration Management (SCM) Plan:

- The SCM process maintains records of the configuration status of all entities that have been placed under configuration control at the project level or higher. This is used with the Change Management process and the Release process of this SCM Plan.
- The records are maintained for the life of the system.
- Reports include the current version, revision, or release status of each CI, a record of changes to the CI since it was placed under configuration control, and the status of problem reports and change requests that affect the CI.

The Acme Interface project uses two tools for configuration management that collect the necessary data to provide status accounting reports.

Serena Version Manager is the focal point of maintaining configuration status. It has the following features:

- Automatically maintains the differences between multiple revisions of the same file
- Automatically assigns revision numbers to files as they are checked in
- Allows revisions of files to be assigned to one of many promotion groups so that the files can be acted upon as a group
- Allows the assignment of version labels. Version labels permanently assign an identifier to particular revisions of a file. This allows for future references to the label to identify the exact revisions of the file even though that file may have changed.
- Requires adding comments for each file. This is used to enter Software Change Request numbers.

The Software Change Request system is used to store the change requests and is available to all team members. Each change request has:

- The research and history of actions involved in the decision to implement that change
- The name of the baseline that includes this change
- A list of the files changed by the request

The features from the tools above are used to generate the following reports for the project team.

- Configuration Items Detailed Status Report
- Configuration Items Change History
- Released Items Report
- Product Baseline Status Report

## **6. Configuration Audits and Reviews**

The term Configuration Audit is defined in Section 1.1 of this document.

This SCM Plan ensures successful completion of functional and physical audits and is used as a prerequisite for the establishment of the product baseline in the initial SCM process. It is used throughout the project for the review of SCM processes and the audit of the configuration repository.

The functional audit determines whether the CI satisfies the functions defined in the specifications. The physical audit determines that all items identified as part of system configuration are present in the product baseline. Sections 6.1 and 6.2 describe how these audits are used.

## 6.1 Functional Configuration Audits

The SUITE Key Terms and Acronyms document defines a **Functional Configuration Audit (FCA)** as, “An inspection to determine whether the (software) configuration item satisfies the functions defined in the specifications. Consists of someone acknowledging having inspected or listed each item to determine it satisfies the functions defined in specifications.”

The Configuration Management Manager (CMM) is responsible for ensuring that a FCA is performed. Results of the completed audit are used by the Configuration Control Board (CCB) to authorize a new baseline. A Functional Audit ensures all requirements are present in the baseline. This audit occurs at the creation of each of the three baselines when the CMM compares the baseline content report, SCR(s), and the Requirements Traceability Matrix. (See the expanded process description, Section 4.4)

The Requirements Traceability Matrix (SEM-0401) includes data required to verify and validate changes against CIs.

The Traceability Matrix contains the following columns:

- Business Requirement Number
- Functional Requirement Number
- System Design Section Number
- Test Reference Number(s)
- Requirement Completion
- Requirement Modification(s) / Remarks

## 6.2 Physical Configuration Audits

The SUITE Key Terms and Acronyms document defines a **Physical Configuration Audit (PCA)** as “the formal examination of the ‘as-built’ configuration of a configuration item against its technical documentation to establish or verify the configuration item's product baseline.”

The CMM is responsible for ensuring that a PCA is performed. Results of the completed audit are used by the CCB to authorize a new baseline.

A physical audit ensures that the correct CI(s) are present in the production baseline. This audit occurs as part of the implementation process when the librarian compares the UAT and Production Baseline Content Reports. (See the expanded process description, Section 4.4)

Physical audits may be requested at any time by the CCB, management or the State’s auditors.

The results of the audit, including any discrepancies are noted in a Service Ticket and entered into the Project Tracker system and assigned to the CMM. The Service Ticket provides documentation of the physical audit.

The CMM notifies the CCB of the results and if no discrepancies are found, the Service Ticket is closed. If discrepancies are found, the ticket remains open until the CMM has resolved the problem and reports the results to the CCB.

## 7. Archive and Retrieval

By using the processes described above software releases are archived automatically. This is done using version labels. As versions of the software are released to each of the three environments (System Test, User Acceptance Testing and Production), the files in the release are given a version label. At anytime in the future a release can be retrieved by using its version label. This allows for any, or all, files in a release to be retrieved. This feature can be utilized in the event that a release needs to be backed out and should be referenced in any back out plans.

The Serena Version Manager tool allows for keeping the difference of changed files and not the complete changed file. This reduces the total disk space requirements as compared to some other repository solutions. Because of this, the files checked into Serena Version Manager are kept indefinitely.

Should circumstances require archival to offline storage, several options are available: Releases/baselines can be extracted from the repository and copied to alternative media such as data-DVD or tape. An empty Serena Version Manager project database can be created and populated with all files from the current release. The old project database could then be copied to offline media to save disk space. Should files be needed from this archive, they may be reloaded to disk and the necessary files extracted.

A major rewrite of the system may necessitate the creation of a new and empty Serena Version Manager project database. Any files needed from the previous release would be added to the repository. This can give the files a new revision number. The old project database could then be archived to off line storage.

## 8. Training

These Software Configuration Management procedures and tool sets are already in place for the Statewide Financial System, therefore no new training is required.

If contract workers are added to augment the staff for this project they will need to be trained on the process and procedures they are expected to follow. It is each State of Michigan manager's responsibility to ensure respective contract workers follow the SCM procedures. Normally this is included in the peer mentoring that is given to any new staff.

## 9. Related Forms

- Requirements Traceability Matrix (SEM-0401)
- Installation Plan (SEM-0702)

## Approval Information

The signatures relay an understanding of the purpose and content of the document by those endorsing it.

☐ Approve ☐ Approve with Modifications ☐ Reject

Comments:

"Click HERE and Type"

### Initial Approval Signatures

	Name / Title	Signature	Date
Client Sponsor			
MDIT Sponsor			

	<b>Name / Title</b>	<b>Signature</b>	<b>Date</b>
Project Manager			
Configuration Management Manager			

**Final Approval Signatures**

	<b>Name / Title</b>	<b>Signature</b>	<b>Date</b>
Client Sponsor			
MDIT Sponsor			
Project Manager			
Configuration Management Manager			